PROBABILISTIC MODEL-BASED SENTIMENT ANALYSIS OF TWITTER MESSAGES

Asli Celikyilmaz[1]* *Dilek Hakkani-Tür*[2]† *Junlan Feng*[3]

[1]University of California, Berkeley
[2]Speech at Microsoft | Microsoft Research, Mountain View, CA
[3]AT&T Labs-Research, Florham Park, NJ

ABSTRACT

We present a machine learning approach to sentiment classification on twitter messages (tweets). We classify each tweet into two categories: polar and non-polar. Tweets with positive or negative sentiment are considered polar. They are considered non-polar otherwise. Sentiment analysis of tweets can potentially benefit different parties, such as consumers and marketing researchers, for obtaining opinions on different products and services. We present methods for text normalization of the noisy tweets and their classification with respect to the polarity. We experiment with a mixture model approach for generation of sentimental words, which are later used as indicator features of the classification model. Based on a gold standard manually annotated ensemble of tweets, with the new approach, we obtain F-scores that are relatively 10% better than a classification baseline that uses raw word n-gram features.

Index Terms—

Sentiment analysis, micro-blogs, feature extraction, probabilistic graphical models, Twitter.

1. INTRODUCTION

Twitter is a *micro-blogging* social networking web site that has a large and rapidly growing user base. The messages on twitter are in the form of *tweets*, which are short status updates (of 140 characters or less). As an increasingly popular platform for conveying opinions and thoughts, it seems natural to mine Twitter for potentially interesting trends regarding prominent topics on any type of news, products and events. A sentiment classification model based on the Twitter data could provide unprecedented utility for different parties for instance for marketing and financial purposes. For example, a business could gauge its recent marketing campaign by aggregating user opinions on Twitter regarding their products.

In general, textual information can be broadly categorized into *facts* and *opinions*. Facts are objective expressions about entities, events and their properties. Opinions are usually subjective expressions that describe people's sentiments, appraisals or feelings toward entities, events and their properties. Although the opinions are generally more broad, here we only focus on opinion expressions that convey people's positive or negative sentiments.

In many previous studies, research on sentiment classification is conducted on review-type data, such as movie or restaurant reviews [1], blogs [2], and news [3]. These data sets often consist of relatively well-formed, coherent and at least paragraph-length pieces of text. Furthermore, resources such as polarity lexicons, and parsers are usually available for these domains. Sentiment analysis on Twitter, however, is different from the sentiment analysis models on reviews or blogs based on machine learning. In a tweet message, a sentiment is conveyed in one or two sentence passages, which are rather informal, including abbreviations and typos. These messages are less consistent in terms of language usage, and usually cover a much wider array of topics. Also, sentiment is not always as obvious when discussing human-generated status updates; many tweets are ambiguous even to a human reader as to their sentiment. Finally, a considerably large fraction of tweets convey no sentiment whatsoever, such as advertisements and links to news articles, which provide some difficulties in data gathering, training and testing. In Table 1, we provide examples of tweets with and without polarity from our data set.

Sentiment analysis of twitter messages has recently received great attention from both research and industry. [4] analyzed over 150,000 twitter posts, and found out that 19% of the posts mention a brand name, and 20% of these contained some expression of brand sentiments, and 50% of these had positive, and 33% were critical of the company or product. They discussed the implications for corporations using microblogging as part of their overall marketing strategy. [5] reported a 2-step automatic sentiment analysis method for twitter using noisy training data. It first classified tweets as subjective(polar) and objective(non-polar), and further distinguished polar tweets as positive and negative. The training data were collected from three popular real-time tweet senti-

*Research supported in part by ONRN00014-02-1-0294, BT Grant CT1080028046, Azerbaijan Ministry of Communications and Information Technology Grant, Azerbaijan University of Azerbaijan Republican the BISC Program of UC Berkeley.

†This work was done while the author was working for the International Computer Science Institute, with funding from AT&T Labs-Research.

Polarity	Text
Neutral	Job911.com SEM Internet Sales Manager at [COMPANY-NAME] (Reno, NV): SEM Internet Sales Manager * Location: Reno, ... http://bit.ly/xy3ts76 More Jobs @
Neutral	Me and Jesus sang "Breathe" and Got Second Place at T.O.B's Talent Show!! :)
Polar	@TmMinjP I Loveeeeee [COMPANY-NAME] !!!!
Polar	finally! my phone works. [COMPANY-NAME] u are lousy. im getting [COMPANY-NAME] asap!

Table 1. Examples of neutral and polar twitter messages. Note that organization names are replaced with the [COMPANY-NAME] token, and all names and web addresses are distorted in these examples.

ment detection web sites. In this paper, we focus on polarity detection only. Most available twitter sentiment analysis web sites such as TweetFeel are not open on how their classifiers were built. As they described at a very high level (at their web site: http://twittersentiment.appspot.com/), word-spotting rules, word unigram and bigram features have been utilized for classification. [6] proposed searching for twitter messages that contain happy and sad emoticons to collect a microblogging sentiment corpus. Then they compare naive Bayes, support vector machine and conditional random fields with word n-gram features for sentiment detection. Similarly, [7] and [8] proposed methods for automatic sentiment detection on blog posts. Finally, [9] investigated use of topic modeling for the analysis of disaster-related twitter data.

In this paper, we focus on polarity detection in tweets, and first propose a new method for text normalization and investigate its effect when used for cleaning tweets for polarity detection. We use pronunciations of words to map alternative and shorter spellings into the intended words. This method reduces the sparseness caused by the noise in tweets. Then, we propose a new method for extracting a polarity lexicon from tweets, and extract a set of features based on this lexicon. We apply commonly used machine learning techniques, such as boosting, for tweet classification to identify sentiments. We argue that, in cases with noisy text, such as the tweets, the use of off-the-shelf polarity lexicons may not be as useful as using lexicons extracted from the given data set.

Section 2 presents the tweet data collection efforts and is followed by our approach to normalizing and cleaning tweet messages to enable learning polarity patterns in section 3. Section 4 describes the generative models we use to extract a group of polarity-related words, which are then used in determining the polarity of tweets. We describe the data sets and evaluation measures and present the related experimental results in section 5 and draw conclusions in section 6.

2. DATA COLLECTION

We collected 2 million tweets using Twitter search API from September 2009 to June 2010. Queries we used were a set of keywords related to mobile operation. Since Twitter API only allows clients to make a limited number of calls in a given hour, we distribute our data collection process into several machines. We only kept unique tweets in our data archive using tweet identification number as the unique key. We stored tweets from the same hour into a text file in the Java Script Object Notation (JSON) format. Each retrieved tweet consists of 18 fields including tweet text, user id, time the tweet is submitted, and many other meta data. We indexed the collected tweets using an off-the-shelf text indexing tool - Lucene- to support us observe the data through search. We used keyword search to identify tweets targeting certain entities, such as products and organizations.

3. TWITTER MESSAGE NORMALIZATION

While the writing style and the lexicon of tweets is widely varied, much of them are similar to SMS text messages. This is likely because many users access Twitter through mobile devices. Posts are often highly ungrammatical, and filled with spelling errors. In order to clean the dataset, we implemented the following steps:

3.1. Text Processing

We captured a set of tweet patterns, which are formulated as regular expressions, for text normalization. The normalized text is then used in the feature extraction process:

- **Numeric Expressions:** The following replacements are used for numeric expressions:
  - monetary amounts, such as $5,943 are replaced with the <DOLLAR−AMOUNT> token,
  - all percentages, such as 50% are replaced with the <PERCENTAGE> token, and
  - all the remaining tokens that include numbers are replaced with the <NUMBER> token.
- **HTML Links:** All tokens starting with "http://", "https://", and "www." are replaced with the <URL> token.
- **User identifiers:** All user names, that are tokens that start with an "@" character are replaced with the <USER> token.
- **Target Organization Names:** Expressions that refer to the entity names that the polarity detection is targeting are replaced with the <ENT> token. To capture several

variations of the target organization names, we again manually formed a set of regular expressions.

3.2. Pronunciation-Based Word Clustering

Most tweets include encrypted words, written in a different way than the intended word, but pronounced similarly, such as "ifoan and "ifone", instead of "iPhone", "2nite" instead of "tonight". We estimate the pronunciation of all the words in the twitter messages by using the lexical lookup tools available in the Festival Speech Synthesis System [10]. Then, we cluster the words that have the same pronunciation, and use the same token for these. Note that, this step is rather preliminary and noisy, as it results in merging homonyms as well. Therefore, in this work, we first mined a list of English homonyms from the internet, and the discarded the clusters formed from homonyms from the set of generated clusters by using these lists.

4. POLARITY LEXICON EXTRACTION VIA MIXTURE MODEL

A probabilistic generative model specifies a stochastic procedure by which data can be generated, usually making reference to unobserved random variables that express latent structure. Statistical inference probability distributions over latent variables are computed (higher order uncertainty approximation) conditioned on a given dataset. These approaches are very useful in statistical natural language processing, especially when words are generated from the latent structure of the intention of the given utterances.

Probabilistic models of language, such as topic models, are typically driven by long-term dependencies between words, i.e., bag-of-words assumptions, which generate documents based on semantic correlations between words, independent of their word order. In this work, we used a topic model based probabilistic approach, namely Latent Dirichlet Allocation (LDA) [11] to capture polarity words generating semantic content of tweet messages. It should be noted that, we use the LDA model to extract semantic concepts as probability distributions over words that tend to co-occur in text. Such terms form meaningful groupings which can then be used to infer different topics being mentioned in text.

Of the particular importance of choosing LDA models over other its most closest counterpart, namely the probabilistic latent semantic analysis (pLSI) is that in pLSI, each document is represented as a list of numbers (mixing proportions for topics), and there is no generative probabilistic model for these numbers. This leads to two problems, (1) the number of parameters grow with the size of the corpus, (2) it is not clear how to assign probability to a document outside the training set. Thus in information retrieval, pLSI may is not usually preferred (see the discussion in [11]). Our aim in using LDA is not so different. We would like to find groupings of polar words in the given tweet messages so they can be used as additional features for our classification model.

4.1. Tweet Message Topic Model

A tweet message is represented as a mixture of fixed topics, with topic z getting weight $\theta_z^{(t)}$ in a tweet t and each topic is a distribution over a finite vocabulary of words, with word v having a probability $\phi_v^{(z)}$ in topic z. Placing symmetric Dirichlet priors on $\theta^{(t)}$ and $\phi^{(z)}$, with $\theta^{(t)} \sim Dirichlet(\alpha)$ and $\phi^{(z)} \sim Dirichlet(\beta)$, where α and β are hyper-parameters to control the sparseness of distributions, the generative model is given by:

$$\begin{array}{rcll} v_i|z_i, \phi_{v_i}^{(z_i)} & \sim & Discrete(\phi^{(z_i)}), & i = 1, ..., V \\ \phi^{(z)} & \sim & Dirichlet(\beta), & z = 1, ..., K \\ z_i|\theta^{(t_i)} & \sim & Discrete(\theta^{(t_i)}), & i = 1, ..., V \\ \theta^{(t)} & \sim & Dirichlet(\alpha), & t = 1, ..., T \end{array} \quad (1)$$

where T is the number of tweets in the collection, K is the total number of topics, V is the total number of words in the tweet collection, and t_i and z_i are the tweet and the topic of the i^{th} word v_i, respectively. Each word in the vocabulary $v_i \in V = \{v_1, ...v_V\}$ is assigned to each latent topic variable $z_{i=1,...,V}$ of words. To calculate the expected posterior probabilities $\hat{\phi}_{v_i}^{(z_i)}$ of a word v_i in a given tweet given a topic $z_i = k$ we use the count matrices:

$$\hat{\phi}_{v_i}^{(z_i)} = \frac{n_{v_i k}^{VK} + \beta}{\sum_{j=1}^{V} n_{v_j k}^{VK} + V\beta} \quad (2)$$

where $n_{v_i k}^{VK}$ is the count of v_i in topic k. We iterated over a set of possible values for hyper-parameters and number of clusters and analyzed the performance change based on the perplexity of the expected posterior mixing proportions.

4.2. Polar Lexicon Extraction

Our aim is to capture words that co-occur together in polar tweets so they could be used as an indicator feature later while building the classification model for identifying tweets that include sentimental phrases. We build the LDA model on the tweet collection without the non-polar tweets. This enables us to extract the polar tweet lexicon. In the clustering experiments, we only used the normalized words as a lexicon to extract polarity lexicon (the information gained from the clustering according to the word pronunciations is used to build the vocabulary of topic modeling.) We iterated the models using $k = \{3, 5, 10\}$, where 5 clusters was found to be the optimum based on perplexity.

We use the K discrete topic probability distributions over words, $\hat{\phi}_{v_i}$, and selected the top 30 words from each with the highest probability, under the assumption that they would represent different polar concepts. These words constitute our

Hidden Topic	*Extracted Unigrams*
k=1	*best, good, slow, kind, cares, wow, hiked, reason*
k=2	*useless, friggin, lied, bitch, restrictions, kill, failure, fail, fg, lying*

Table 2. Sample words in polar lexicon with high probability in LDA topic-word distributions. Two groups indicate words from two different topics.

polar lexicon. Each polar vocabulary is used as unigram feature for the sentiment classification model. We compiled a polar lexicon, with scores assigned to words, using their probabilities. A sample list of words in the lexicon is shown in Table 2. These lists include tokens that are typical of tweets, and may not be seen in other textual genre, such as 'fg'.

5. SENTIMENT CLASSIFICATION

5.1. Data Sets

Experiments were performed using two data sets, manually annotated with polarity (positive, negative, and none). The positive and negative examples in both data sets were grouped to form the examples with polarity. The first set was used to train and test classification experiments with n-fold cross validation, where a single message was used as the test set in each fold, and then computing the evaluation measures over all the examples. The first set is also used to extract the polarity lexicon using the approach described in section 4. The second set is used only as a test set. The number of examples, the percentage of polar messages and the average number of tokens in the two data sets after data cleaning are presented in Table 3.

Both data sets are randomly selected from the tweet archive described in section 3.

	Set-1	**Set-2**
No. messages	1,064	1,004
Perc. of polar messages	9.1%	11.2%
Avg. No. Tokens/Message	15.8	15.8

Table 3. Data set statistics.

5.2. Evaluation Measures

For benchmark analysis we used two measurements: error rate and F-measure. Error rate is the percentage of examples that are assigned the wrong category:

$$\textbf{Error-rate} = 100\% - accuracy \quad (3)$$

and F-measure is the harmonic mean of recall and precision, where recall is the ratio of polar examples that are found by a method, and precision is the percentage of correctly found polar examples amongst the ones that a polarity detection system marked as polar:

$$\textbf{F-measure} = 2 * \frac{precision * recall}{precision + recall} \quad (4)$$

5.3. Impact of Word Clustering for Text Normalization on Classification

As described in section 3.2, the words in the two data sets are clustered using their pronunciations. As a result, 695 words were grouped into 281 clusters. Some example groups are: "available, availible", and "right, rite". We manually examined the formed clusters for correctness, and found that about 30% of the clusters were formed from homonyms and removed them from this set. After this manual cleaning step, for the first data set, a word token was replaced with the corresponding cluster for 69.5% of the messages. This ratio is 77.7% for the second data set. Hence, even though the number of clusters does not seem large, most messages are affected by this normalization.

To test the effect of using pronunciations for text normalization of twitter messages, we performed polarity detection experiments using the BoosTexter [12] classifier with all word unigrams, bigrams, and, trigrams as features. BoosTexter is a general purpose machine-learning program based on boosting for building a classifier from text data, which can handle multi-class problems such as unbalances in different classes of the data (the polarity tweets compared to the rest of the tweets have fairly small proportion) as well as problems in which some instances belong to more than one class. The results of these experiments are listed in Table 4. In experiments with the test set Set-1, we trained boosting with 1,064-fold cross validation, and used one example as the test example in each fold. In the experiments with Set-2, we trained the classifier using Set-1, and tested the resulting models on Set-2. The baseline row corresponds to experiments where each example is assigned the majority class (non-polar in this case). The clustering of the words with the same pronunciation helped reduce error rates and improve F-measure on both data sets (by close to 10% relative on Set-1, and slightly on Set-2).

5.4. Classification Results with Polarity Lexicons

In section 4, we automatically extracted a polar lexicon based on probabilistic models, i.e., LDA. The normalized text was used in this process, resulting in a polarity terms lexicon of 133 words. Using the top N words from the extracted lexicon

	Set-1 (n-fold)				Set-2			
	Error bf %	**Precision** %	**Recall** %	**F-meas** %	**Error** %	**Precision** %	**Recall** %	**F-meas** %
Baseline	9.1	-	-	-	11.2	-	-	-
No clustering	8.2	58.9	34.0	43.1	10.6	55.7	25.9	35.4
With clustering	**7.6**	62.7	38.5	**47.7**	**10.5**	56.9	25.9	**35.6**

Table 4. Error rate and F-measure resulting from the classification experiments. In these experiments, all word unigrams, bigrams and trigrams are used as features.

(sorted according to expected posterior topic-word probabilities of words in each topic), we built a discriminative model, a classifier engine to predict the polarity of a tweet given the polar words. Similar to the experiments of the previous section, we built two BoosTexter classifier models using the unigrams of polar words as indicator features: (1) first model uses Set-1 dataset and builds the classifier based on n-fold cross validation, (2) second models uses Set-1 for training and Set-2 for testing purposes. The error rates are shown in Table 5. In these experiments, the polarity lexicon is extracted from Set-1, hence the lexicon is biased towards the test set, and Set-2 is the previously unseen test set. The classification performance on Set-1 remains constant or degrades as more polar word tokens are used in classification. This is mainly because of the decrease in precision due to over-training as more features are used. However, on the unseen test set, the classification performance improves, and reaches the peak when using the unigram features formed from the top 100 tokens from the probabilistic lexicon. The F-measure at this point is more than 10% relatively better than the classification approach that uses text normalization and all word unigram, bigram and trigrams as features.

An additional list of polarity words was available in the MPQA opinion corpus, as part of the subjectivity lexicon [13]. This corpus contains news articles and other text documents manually annotated for opinions and other private states (i.e., beliefs, emotions, sentiments, speculations, etc.). The MPQA list of polarity words contains 8,221 word tokens of positive and negative polarity words. In order to compare the performance of the extraction of the polarity lexicon extraction, we tested the same approach using the publicly available MPQA list. The results are shown on the last row of Table 5. As seen, the off-the shelf polarity lexicon is not as useful as the lexicon automatically built from noisy twitter messages due to various reasons. First, the MPQA lexicon is tuned for the cleaner text genre, and it contains a large set of tokens without any appropriateness or precedence scores for the specific domain, resulting in a large set of features for classification (in comparison to the estimated polarity lexicons). However, the poorer performance is not as severe on the second set, suggesting that it is robust.

6. CONCLUSIONS

We have shown that in the very noisy domain of customer feedback, it is nevertheless possible to perform sentiment classification. This can be achieved by using various machine learning methods to first normalize noisy tweets and then by implementing a mixture model component on polar tweets to capture polar vocabularies for a discriminative learning approach. The experimental results indicate that for tweet messages it is possible to capture the sentimental messages posted by humans (not automatically generated advertisements) based on using the best features of discriminative and generative learning models. We were able to demonstrate that it is not necessary to exploit a large dictionary of polar words to capture the sentimental utterances in everyday human conversations. A natural way would be generating the lexicon from the already seen unlabeled tweet messages. As a future work, we plan to build a generative model around entities to learn the lexicon that can be extracted from tweets so as to represent the similar concepts such as *love*, *lovwww*, *loveee* and *luv* as one entity 'love'. An important future direction lies in augmenting an entity-based concept model with lexico-semantic knowledge. We plan to cluster entities based on semantic relatedness. This model could enable automatic attribution of the latent concepts/topics defined as a distribution over entities, which are naturally formed based on co-occurrence of their most overlapping attributes.

Acknowledgments: The authors would like to thank the anonymous reviewers for their valuable comments.

7. REFERENCES

[1] A. Kennedy and D. Inkpen, "Sentiment classification of movie reviews using contextual valence shifters," *Computational Intelligence*, vol. 22, no. 2, pp. 110–125, 2006.

[2] P. Melville, W. Gryc, and R. Lawrence, "Sentiment analysis of blogs by combining lexical knowledge with text classification," in *in Proceedings of the 15th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*, 2009.

	Set-1 (n-fold)				Set-2			
	Error bf %	**Precision %**	**Recall %**	**F-meas %**	**Error %**	**Precision %**	**Recall %**	**F-meas %**
Top 20	**6.5**	91.2	31.9	**47.4**	9.7	89.4	15.2	26.0
Top 30	6.6	90.9	30.9	46.1	9.5	90.5	17.0	28.6
Top 50	**6.5**	96.7	29.9	45.7	9.4	82.1	20.5	32.8
Top 100	6.7	93.3	28.9	44.1	**9.1**	76.9	26.8	**39.7**
Top 133	6.7	93.3	28.9	44.1	9.8	63.8	26.8	37.7
MPQA	10.2	36.4	16.5	22.7	9.8	68.6	21.4	32.6

Table 5. Error rate and F-measure resulting from classification experiments, where only the polarity words are used as features.

[3] N. Godbole, M. Srinivasaiah, and S. Skiena, "Large-scale sentiment analysis for news and blogs," in *Proceedings of the International Conference on Weblogs and Social Media (ICWSM)*, 2007.

[4] B. J. Jensen, M. Zhang, K. Sobel, and A. Chowdury, "Twitter power: Tweets as electronic word of mouth," *Journal of the American Society for Information Science and Technology*, vol. 60, no. 11, pp. 2169–2188, 2009.

[5] L. Barbosa and J. Feng, "Robust sentiment detection on twitter from biased and noisy data," in *Proceedings of the 23rd International Conference on Computational Linguistics (COLING)*, 2010.

[6] A. Pak and P. Paroubek, "Twitter as a corpus for sentiment analysis and opinion mining," in *Proceedings of LREC*, 2010.

[7] F. Liu, D. Wang, B. Li, and Y. Liu, "Improving blog polarity classification via topic analysis and domain adaptation," in *Proceedings of NAACL-HLT*, 2010.

[8] J. Kim, J.-J. Li, and J.-H. Lee, "Discovering discriminative views: Measuring term weights for sentiment analysis," in *Proceedings of ACL-IJCNLP*, 2009.

[9] A. Kireyev, L. Palen, and K. M. Anderson, "Applications of topic models to analysis of disaaster-related twitter data," in *Proceedings of NIPS Workshop on Applications for Topic Models: Text and Beyond*, 2009.

[10] A. W. Black, P. Taylor, and R. Caley, "The festival speech synthesis system system documentation edition 1.4, for festival version 1.4.3," Dec 2002.

[11] D. M. Blei, A. Ng, and M. Jordan, "Latent dirichlet allocation," in *Jrnl. Machine Learning Research, 3:993-1022*, 2003.

[12] R. E. Schapire and Y. Singer, "BoosTexter: A boosting-based system for text categorization," *Machine Learning*, vol. 39, no. 2/3, pp. 135–168, 2000.

[13] T. Wilson, J. Wiebe, and P. Hoffmann, "Recognizing contextual polarity in phrase-level sentiment analysis," in *in Proc. of HLT-EMNLP*, 2005.